University of Glasgow

School of Computing Science

60 YEARS OF COMPUTING AT GLASGOW

Transport Security

Networked Systems (H)

Lecture 17

Lecture Outline

- Pervasive traffic monitoring
- Confidentiality and authentication
- Securing network transport

Pervasise Monitoring

- Possible to intercept traffic on a network
- Many countries monitor traffic, for legal reasons
    - To enable authorised wiretaps by the police, for example
    - Much of this is desirable – the *are* good reasons why law enforcement need to intercept *some* traffic
    - Edward Snowden revelations show *pervasive monitoring* is widespread
        - IETF consensus is that “we cannot defend against the most nefarious actors while allowing monitoring by other actors no matter how benevolent some might consider them to be, since the actions required of the attacker are indistinguishable from other attacks” – RFC 7258 “Pervasive Monitoring is an Attack” (https://tools.ietf.org/html/rfc7258)

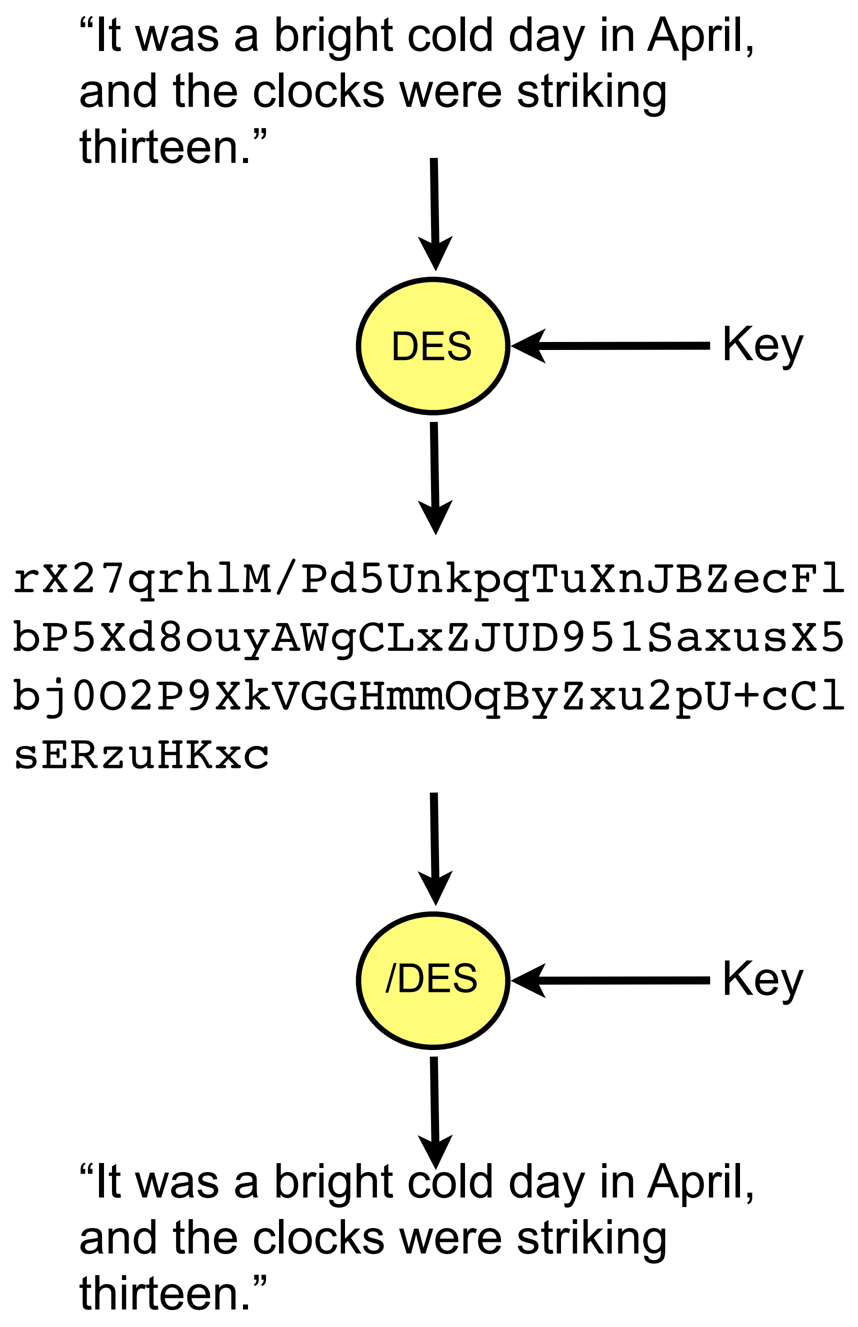
Edward Snowden

Confidentiality

- Must encrypt data to achieve confidentiality
- Two basic approaches
  - Symmetric cryptography
    - Advanced Encryption Standard (AES)
  - Public key cryptography
    - The Diffie-Hellman algorithm
    - The Rivest-Shamir-Adleman (RSA) algorithm
    - Elliptic curve-based algorithms
  - Complex mathematics – will not attempt to describe

Symmetric Cryptography

- Function converts plain text into cipher-text
  - Fast – suitable for bulk encryption
  - Cipher-text is binary data, and may need base64 encoding
- Conversation is protected by a secret key
  - The same key is used to encrypt as is used to decrypt
  - Key must be kept secret, else security lost – a problem: how to distribute the key?



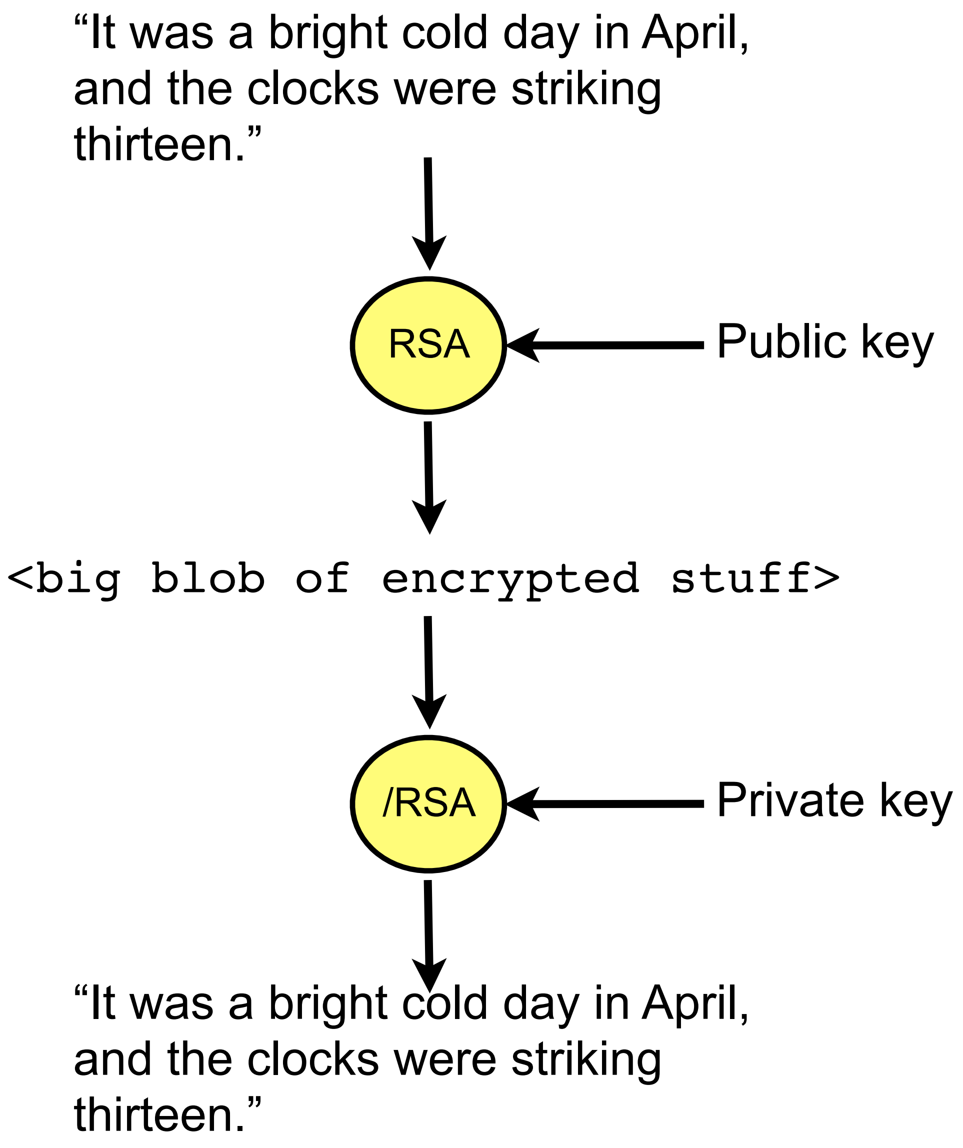
"It was a bright cold day in April, and the clocks were striking thirteen."

```
rX27qrhlM/Pd5UnkpqTuXnJBZecFl
bP5Xd8ouyAWgCLxZJUD951SaxusX5
bj0O2P9XkVGGHmmOqByZxu2pU+cCl
sERzuHKxc
```

"It was a bright cold day in April, and the clocks were striking thirteen."

Public Key Cryptography

- Key split into two parts:
  - Public key – is widely distributed
  - Private key – must be kept secret
- Encrypt using public key → need private key to decrypt
  - Public keys are published in a well known directory → solves the key distribution problem
  - Problem: very slow to encrypt and decrypt


"It was a bright cold day in April, and the clocks were striking thirteen."
RSA
Public key
<big blob of encrypted stuff>
/RSA
Private key
"It was a bright cold day in April, and the clocks were striking thirteen."

Hybrid Cryptography

- Use combination of public-key and symmetric cryptography for security and performance
  - Generate a random, ephemeral, *session key* that can be used with symmetric cryptography
  - Use a public-key system to securely distribute this session key – relatively fast, since session key is small
  - Encrypt the data using symmetric cryptography, keyed by the session key
  - Examples: PGP for email, Transport Layer Security (TLS) for web pages

Authentication

- Encryption can ensure confidentiality – but how to tell if a message has been tampered with?
  - Use combination of a *cryptographic hash* and public key cryptography to produce a *digital signature*
  - Gives some confidence that there is no *man-in-the-middle* attack in progress
- Can also be used to prove origin of data

Cryptographic Hash Functions

- Generate a fixed length (e.g., 160 bit) hash code of an arbitrary length input value
  - Should not be feasible to derive input value from hash
  - Should not be feasible to generate a message with the same hash as another
- Examples:
  - MD5 and SHA-1 (weaknesses found in both – care required!)
  - SHA-256

SHA256(“It was a bright cold day in April, and the clocks were striking thirteen”)
= 0fc5c1f4082e697b211cdfa12479b4b3dd57c8da69c8904f5e0fc32499cf4245

Digital Signature Algorithms

- Generating a digital signature:
  - Generate a cryptographic hash of the data
  - Encrypt the hash with your *private key* to give a *digital signature*
- Verifying a digital signature:
  - Re-calculate the cryptographic hash of the data
  - Decrypt the signature using the public key, compare with the calculated hash value → should match

Existing Secure Protocols

- Existing security protocols give confidentiality and authentication:
  - IPSec – useful for VPNs
  - Secure Sockets Layer (SSL) – obsolete, use TLS instead
  - Transport Layer Security (TLS) – general purpose security for TCP-based applications
  - Datagram TLS – for securing UDP-based applications
  - Secure RTP – for securing interactive multimedia applications
  - Secure shell (ssh) – for securing remote login applications
- Use them – don't try to invent your own!

Using TLS

- IETF provides guidelines for how best to use TLS: https://tools.ietf.org/html/rfc7525
    - Read this if you use TLS in your application – and check for updates first
    - IETF “Using TLS in Applications” working group https://datatracker.ietf.org/wg/uta/charter/

- State-of-the-art in TLS implementations is in flux
    - OpenSSL is popular, but poor quality
    - Alternatives in rapid development as of early 2017 – not clear which is the best long term option

Key Escrow

Keys Under Doormats:

MANDATING INSECURITY BY REQUIRING GOVERNMENT ACCESS TO ALL DATA AND COMMUNICATIONS

Harold Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matthew Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Peter G. Neumann, Susan Landau, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Michael Specter, Daniel J. Weitzner

Abstract

Twenty years ago, law enforcement organizations lobbied to require data and communication services to engineer their products to guarantee law enforcement access to all data. After lengthy debate and vigorous predictions of enforcement channels "going dark," these attempts to regulate the emerging Internet were abandoned. In the intervening years, innovation on the Internet flourished, and law enforcement agencies found new and more effective means of accessing vastly larger quantities of data. Today we are again hearing calls for regulation to mandate the provision of exceptional access mechanisms. In this report, a group of computer scientists and security experts, many of whom participated in a 1997 study of these same topics, has convened to explore the likely effects of imposing extraordinary access mandates.

We have found that the damage that could be caused by law enforcement exceptional access requirements would be even greater today than it would have been 20 years ago. In the wake of the growing economic and social cost of the fundamental insecurity of today's Internet environment, any proposals that alter the security dynamics online should be approached with caution. Exceptional access would force Internet system developers to reverse "forward secrecy" design practices that seek to minimize the impact on user privacy when systems are breached. The complexity of today's Internet environment, with millions of apps and globally connected services, means that new law enforcement requirements are likely to introduce unanticipated, hard to detect security flaws. Beyond these and other technical vulnerabilities, the prospect of globally deployed exceptional access systems raises difficult problems about how such an environment would be governed and how to ensure that such systems would respect human rights and the rule of law.

- Effective security is difficult – failures tend to be due to bad implementations or protocols, not weak crypto
- So-called exceptional access or key escrow systems will be discovered, and exploited, by malicious actors – we do not have the expertise to secure such systems
- Design systems to limit access to keying material

H. Abelson, *et al.*, "Keys under doormats: Mandating insecurity by requiring government access to all data and communications", MIT Computer Science and Artificial Intelligence Lab, technical report MIT-CSAIL-TR-2015-026, July 2015. http://dspace.mit.edu/handle/1721.1/97690

Summary

- Pervasive monitoring
- Confidentiality, authentication, and crypto
- Secure transport protocols
- Key escrow